Attorney Docket No.: SSL1P001/SS-010

PATENT APPLICATION

METHOD AND SYSTEM FOR RESTRICTING USE OF A CLIPBOARD APPLICATION

Inventor: Patrick Zuili
696 Towle Way, Apt. 39
Palo Alto, CA 94306
USA
Citizenship: France

Assignee: SecretSEAL Inc.

Express Mail Label #ET459241957 US;
Date of Deposit: 12/21/2001

I hereby certify that this paper or fee is being deposited with the United States Postal Service using "Express Mail Post Office To Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to "Assistant Commissioner for Patents, Washington, DC 20231."
Name: Joe Zheng
Signature:

BEYER WEAVER & THOMAS, LLP
Telephone (650) 961-8300

METHOD AND SYSTEM FOR RESTRICTING USE OF A CLIPBOARD APPLICATION

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] The present invention relates generally to restricting access to documents on computers. More particularly, the present invention relates to a method and system for preventing unauthorized copying of content from a secured document in one application program to another application program.

2. Description of the Related Art

[0002] The Internet is the fastest growing telecommunications medium in history. The growth and the easy access it affords have significantly enhanced the opportunity to use advanced information technology for both the public and private sectors. The Internet also provides unprecedented opportunities for interaction and data sharing among businesses and individuals.

[0003] Unfortunately, however, the advantages provided by the Internet come with a significantly greater element of risk to the ownership, copyright, piracy, security and integrity of information. For example, one can easily copy an article from a web site into his own computer, and then use the unauthorized copy for his own purposes. In particular, anyone using a web browser application to view an article posted at a web site on the Internet can easily copy some or all of the article by simply highlighting the desired portion of the article, and then selecting a "copy" command made available by the web browser application. The copied text can thereafter be literally used for any purpose. Such unauthorized copying of information or content presents serious concerns for electronic publishers and others.

[0004] Furthermore, numerous proprietary and sensitive documents are created daily in a wide range of industries. Although measures may be taken to secure these documents, such as password protection or encryption, once these documents are displayed by a web browser or other application, one can normally perform a "copy" command to copy some or all of a document being displayed into

a designated region in memory referred to as a buffer. In Microsoft Windows applications, the buffer is referred to as a clipboard which is provided by a clipboard application that is part of the Windows operating system. A "copy" command stores selected content to the clipboard and a "paste" command retrieves content from the clipboard. Once content has been copied to the clipboard, it remains in the clipboard until replaced by a different set of content. Hence, content in the clipboard can be repeatedly copied to as many different documents as the user desires.

[0005] One of the features of the clipboard is that it can be accessed by most application program running on the computer. As a result, the user is not limited to inserting the copied content into the same document from which it was copied from. Rather, the content stored in the clipboard can be placed within any document displayed by the computer. For example, the user can select a portion of text from a document created with a word processing application and copy it to the clipboard, switch to another text document created with the same or different word processing program, and paste it into that document from the clipboard. In the same manner, the user can copy the content from the clipboard into a non-text document, such as a drawing document created with a graphics application.

[0006] Thus, there is a need for techniques or mechanisms to control clipboard usage when content from a secured document is being copied.

SUMMARY OF THE INVENTION

[0007] Broadly speaking, the invention relates to techniques or mechanisms for controlling copying of content from a secured file or secured document. In one embodiment, the techniques or mechanisms operate to control clipboard usage such that content from a secured document of one application is not able to be copied to another application or a different document of another application by way of a clipboard. According to another embodiment, alternate content can be copied to another application or a different document of another application instead of the content from the secured document.

[0008] The invention can be implemented in numerous ways including, a method, system, device, and a computer readable medium. Several embodiments of the invention are discussed below.

[0009] As a computer-implemented method for restricting use of a clipboard application in a multi-application computing environment, one embodiment of the invention includes at least the acts of: receiving a copy selection associated with designated content of a source file being displayed by a source application; determining whether the source file is a secured file; and preventing subsequent usage of the designated content in a destination application via the clipboard application when the determining determines that the source file is a secured file.

[0010] As a computer-implemented method for restricting use of a clipboard application in a multi-application computing environment, another embodiment of the invention includes at least the acts of: receiving a copy selection associated with designated content of a source file being displayed by a source application; determining whether the source file is a secured file; and preventing storage of the designated content to the clipboard application when the determining determines that the source file is a secured file.

[0011] As a computer-implemented method for restricting use of a clipboard application in a multi-application computing environment, still another embodiment of the invention includes at least the acts of: receiving a copy selection associated with designated content of a source file being displayed by a source application; initially storing the designated content to the clipboard application; subsequently determining whether the source file is a secured file; and replacing the designated content stored in the clipboard application with alternate content when the determining determines that the source file is a secured file.

[0012] As a computer-implemented method for restricting use of a clipboard application in a multi-application computing environment, yet another embodiment of the invention includes at least the acts of: launching a first application when a request to access a file is received; determining, in an operating system supporting the multi-application computing environment, whether the file being requested is secured; and loading the file in clear mode into the first application while activating a clipboard security monitor when the file is determined to be secured, wherein the

clipboard security monitor ensures that no contents in the secured file can be copied into a second application.

[0013] As a computer readable medium including at least computer program code for restricting use of a clipboard application in a multi-application computing environment, one embodiment of the invention includes at least: computer program code for receiving a copy selection associated with designated content of a source file being displayed by a source application; computer program code for determining whether the source file is a secured file; and computer program code for preventing subsequent usage of the designated content in a destination application via the clipboard application when it is determined that the source file is a secured file.

[0014] Other aspects and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying drawings which illustrate, by way of example, the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[0015] The invention will be readily understood by the following detailed description in conjunction with the accompanying drawings, wherein like reference numerals designate like structural elements, and in which:

FIG. 1 is a flow diagram of copy control processing according to one embodiment of the invention.

FIG. 2 is a flow diagram of clipboard copy processing according to one embodiment of the invention.

FIG. 3 is a flow diagram of clipboard copy processing according to another embodiment of the invention.

FIG. 4 is a flow diagram of clipboard paste processing according to one embodiment of the invention.

FIG. 5 illustrates a first representative copy operation according to one embodiment of the invention.

FIG. 6 is a block diagram of a second representative copy operation according to another embodiment of the invention.

FIG. 7A is a flow diagram of clipboard copy processing according to another embodiment of the invention.

FIG. 7B is a flow diagram of clipboard paste processing according to another embodiment of the invention.

FIG. 8 is a block diagram of a third representative copy and paste operation according to one embodiment of the invention.

FIG. 9 illustrates a block diagram of a clipboard usage control system according to one embodiment of the invention.

FIG. 10 shows some of the possible outcomes of a copy and paste operation from a secured document.

DETAILED DESCRIPTION OF THE INVENTION

[0016] The invention relates to techniques or mechanisms for controlling copying of content from a secured file or secured document. In one embodiment, the techniques or mechanisms operate to control clipboard usage such that content from a secured document of one application is not able to be copied to another application or a different document of another application by way of a clipboard. According to another embodiment, alternate content can be copied to another application or a different document of another application instead of the content from the secured document.

[0017] As used herein, a secured file or secured document can be considered to be interchangeable, though a secured document is one type of secured file. Both secured file and secured document pertain to a type of electronic data that includes, but is not limited to, various types of documents, multimedia files, data, executable codes, images and texts. The secure nature of the files or documents is such that the data cannot be accessed without *a priori* knowledge. One example of the *a priori* knowledge is a password. Another example of the *a priori* knowledge is a file key available only to an authenticated user.

[0018] Embodiments of this aspect of the invention are discussed below with reference to FIGs. 1 - 10. However, those skilled in the art will readily appreciate that the detailed description given herein with respect to these figures is for explanatory purposes as the invention extends beyond these limited embodiments.

[0019] FIG. 1 is a flow diagram of copy control processing 100 according to one embodiment of the invention. The copy control processing 100 operates to control the storage of content to a clipboard. Here, the clipboard pertains to an application clipboard provided by a software application which operates to store content to memory and allows such content to be later retrieved and utilized (e.g., pasted) in another document of the same application or another application.

[0020] The copy control processing 100 initially selects 102 content to be copied. The content to be copied can be selected 102 in a variety of different ways. For example, in one embodiment, the content to be copied can be selected by using a pointing device to highlight the content to be copied. Typically, the selected content represents a portion of content being displayed on a display screen of a computing device. However, the selected content can also pertain to an entire document.

[0021] Next, a decision 104 determines whether the selected content is associated with a secured file. When the decision 104 determines that the selected content is associated with a secured file, then alternate content is stored 106 to the clipboard. Here, because the selected content is associated with a secured file, the selected content is not permitted to be copied to the clipboard. Alternatively, if the selected content was initially stored to the clipboard, the selected content would be removed from the clipboard. In either case, the alternate content is stored 106 to the clipboard.

[0022] The alternate content can take various forms and will typically vary with implementation. As one example, the alternate content can simply be blank (i.e., no content). As another example, the alternate content can be a predetermined message, such as a message that secured content cannot be copied. Still another example of alternate content is to scramble the selected content such that it is not user-discernable.

[0023] On the other hand, when the decision 104 determines that the selected content is not associated with a secured file, then the selected content is stored 108 to the clipboard. Here, the clipboard essentially operates in a conventional fashion, such that the selected content is stored to the clipboard and thus may be subsequently retrieved and utilized elsewhere. Following the operations 106 and 108, the copy control processing 100 is complete and ends.

[0024] FIG. 2 is a flow diagram of clipboard copy processing 200 according to one embodiment of the invention. The clipboard copy processing 200 operates to store content to a clipboard application when a copy command is activated.

[0025] The clipboard copy processing 200 initially opens 202 a first application program. A source file is then retrieved and displayed 204 using the first application program. Next, at least a portion of the source file is selected 206. Here, the source file is displayed using the first application program and thus a user can interact with the first application program via its Graphical User Interface (GUI) to select at least a portion of the source file that is to be copied. Thereafter, the clipboard copy processing 200 determines 208 whether a copy command has been initiated.

[0026] When the decision 208 determines that a copy command has not been received, then other processing 210 can potentially be performed. The other processing 210 can perform a wide variety of other actions (e.g., commands supported by the first application program). Following the other processing 210, if any, the clipboard copy processing 200 returns to repeat the decision 208.

[0027] On the other hand, when the decision 208 determines that a copy command has been received, then a decision 212 determines whether the source file is a secured file. When the decision 212 determines that the source file is a secured file, then alternate content is stored 214 to a clipboard buffer. The clipboard buffer represents a storage area associated with a clipboard application. As noted above with respect to FIG. 1, the alternate content can take a variety of different forms. In addition, the clipboard copy processing 200 could support usage of a plurality of different alternate contents. In such case, the clipboard copy processing 200 could first identify or receive a selection pertaining to one of the available alternate contents, and then proceed to store the selected or identified alternate content to the clipboard buffer. Alternatively, when the decision 212 determines that the source file is not a secured file, then the selected portion of the source file is stored 216 to the clipboard buffer. Following the operations 214 and 216, the clipboard copy processing 200 is complete and ends.

[0028] FIG. 3 is a flow diagram of clipboard copy processing 300 according to another embodiment of the invention. The clipboard copy processing 300 operates to store content to a clipboard application when a copy command is selected.

[0029] The clipboard copy processing 300 initially opens 302 a first application program. A source file is then retrieved and displayed 304 using the first application program. Next, at least a portion of the source file is selected 306. Here, the source file is displayed using the first application program and thus a user can interact with the first application program via its GUI to select at least a portion of the source file that is to be copied.

[0030] Next, a decision 308 determines whether a copy command has been received. At this point a user can interact with the operating system or the first application program to copy the selected portion of the source file. On the other hand, other commands, operations or actions can also be performed at this time. Hence, when the decision 308 determines that a copy command has not been received, then other processing 310 can potentially be performed. Following the other processing 310, if any, the clipboard copy processing 300 returns to repeat the decision 308.

[0031] Once the decision 308 determines that a copy command has been received, then the selected portion of the source file is stored 312 to a clipboard buffer. The clipboard buffer is a memory resource that is allocated to the clipboard application. Next, a decision 314 determines whether the source file is a secured file. When the decision 314 determines that the source file is a secured file, then the selected portion of the source file that has been stored in the clipboard buffer is replaced 316 with alternate content. As previously noted, the alternate content can take a variety of different forms or formats. Following the operation 316, as well as directly following the decision 314 when the source file is not a secured file, the clipboard copy processing 300 is complete and ends. Hence, the clipboard copy processing 300 serves to operate a clipboard application such that secured content is not able to be copied to unsecure applications or documents.

[0032] FIG. 4 is a flow diagram of clipboard paste processing 400 according to one embodiment of the invention. The clipboard paste processing 400 pertains to processing associated with a paste command. The clipboard paste processing 400 typically follows after the clipboard copy processing 200 illustrated in FIG. 2 or the clipboard copy processing 300 illustrated in FIG. 3.

[0033] The clipboard paste processing 400 initially opens 402 a second application program. The second application program can also be referred to as a

destination application program. Next, a decision 404 determines whether a paste command has been received. When the decision 404 determines that a paste command has not yet been received, other processing 406 can potentially be performed. Following the other processing 406, if any, the clipboard paste processing 400 returns to repeat the decision 404 and subsequent blocks. On the other hand, when the decision 404 determines that a paste command has been received, then the content from the clipboard buffer is pasted 408 into the second application program.

[0034] FIG. 5 illustrates a first representative copy operation 500 according to one embodiment of the invention. The first representative copy operation 500, for example, pertains to the clipboard copy processing 200 illustrated in FIG. 2 and the clipboard paste processing 400 illustrated in FIG. 4.

[0035] The procedures performed by the first representative copy operation 500 is as follows. First, a source application 502 is activated such that selected content 504 can be identified (e.g., selected). After the selected content 504 has been identified, a copy button 506 can be activated (e.g., by a button press). The copy button 506 is typically a GUI component displayed on a display screen that is "pressed" (i.e., clicked-on) through use of a pointing device. After the copy button 506 is activated, the selected content 504 is sent to a clipboard security monitor 508. At the clipboard security monitor 508, a determination is made as to whether or not the selected content is permitted to be stored to a clipboard buffer 510. In this regard, the clipboard security monitor 508 can examine information pertaining to the source application 502 or a source file having the selected content 504 therein. In any case, when the clipboard security monitor 508 determines that the source application or the source file is a secured item, then the selected content 504 is not permitted to be stored to the clipboard buffer 510. On the other hand, when the source application and the source file are not secured items, then the selected content 504 can be stored to the clipboard buffer 510. Further, in the case in which the selected content 504 is not permitted to be stored to the clipboard buffer 510, alternate content can instead be stored to the clipboard buffer 510.

[0036] Subsequently, at a destination application wherein the previously selected content is to be reused, a paste button 514 is activated (e.g., by a button

press). Then, the destination application 512 requests and receives (i.e., retrieves) the content previously stored to the clipboard buffer 510. The destination application 512 can then insert the retrieved content into a document or file being displayed by the destination application 512. Hence, in the case in which alternate content was instead stored to the clipboard buffer 510, the retrieved content provided to the destination application 512 is merely the alternate content and not the selected content 504. Consequently, according to the first representative copy operation 500, secured items cannot be copied to and reused from a clipboard buffer.

[0037] FIG. 6 is a block diagram of a second representative copy operation 600 according to another embodiment of the invention. The second representative copy operation 600, for example, pertains to the clipboard copy processing 300 illustrated in FIG. 3 and the clipboard paste processing 400 illustrated in FIG. 4.

[0038] The procedures performed by the second representative copy operation 600 are as follows. A source application 602 initially displays a source file from which selected content 604 is identified (selected). Then, a copy button 606 can be activated to cause the selected content 604 to be delivered and stored in a clipboard buffer 610. Typically, the clipboard buffer is memory storage associated with a clipboard application. A clipboard security monitor 608 detects (or is notified of) the storage of the selected content 604 to the clipboard buffer 610. Thereafter, the clipboard security monitor 608 can interact with the source application 602 to obtain additional information about the source application, such as whether the source application or source file is a secured item. In any case, the clipboard security monitor 608 determines whether the selected content 604 is a secured item. When the clipboard security monitor 608 determines that the selected content 604 is a secured item, then the clipboard security monitor 608 operates to clear the selected content 604 from the clipboard buffer 610. In such case, alternate content can, if desired, be stored to the clipboard buffer 610. Hence, the second representative copy operation 600 retains the selected content 604 in the clipboard buffer 610 only when the source application and/or source file are not secured items.

[0039] Then, the destination application 612 requests and receives (i.e., retrieves) the content previously stored to the clipboard buffer 610. The destination

application 612 can then insert the retrieved content into a document or file being displayed by the destination application 612. Hence, in the case in which alternate content was instead stored to the clipboard buffer 610, the retrieved content provided to the destination application 612 is merely the alternate content and not the selected content 604. Consequently, according to the second representative copy operation 600, secured items cannot be reused from a clipboard buffer.

[0040] As noted above, copying content from one application to another typically involves a copy command and then a subsequent paste command. In the embodiments discussed above with respect to FIGs. 2 and 3, and the representative examples provided in FIGs. 5 and 6, the usage of the clipboard for secured content is controlled on the front end, that is, during or soon following the copy command. However, the control over the usage of the selected content can also be controlled on the back end, that is, during the paste command. Other such embodiments of the invention are discussed below with reference to FIGs. 7A, 7B and 8.

[0041] FIG. 7A is a flow diagram of clipboard copy processing 700 according to another embodiment of the invention. The clipboard copy processing 300 operates to store content to a clipboard application when a copy command is selected.

[0042] The clipboard copy processing 700 initially opens 702 a first application program. A source file is then received and displayed 704 using the first application program. Then, at least a portion of the source file is selected 706. The operations 702-706 are similar to the operations 202-206 and 302-306 discussed above in FIGs. 2 and 3, respectively.

[0043] Following the operation 706, a decision 708 determines whether a copy command has been received. When the decision 708 determines that a copy command has not been received, then other processing 710 is potentially performed. Once the other processing 710, if any, is performed, the clipboard copy processing 700 returns to repeat the decision 708. Once the decision 708 determines that a copy command has been received, then the selected portion of the source file is stored 712 to the clipboard buffer. Hence, in this embodiment, the clipboard copy processing 700 operates to store the selected portion of the source file to the clipboard buffer without regard to whether the source file or the first

application program are of a secure nature. However, if desired, a flag or other indicator can also be stored in the clipboard buffer or elsewhere to indicate whether the selected content stored in the clipboard buffer is a secured item.

[0044] FIG. 7B is a flow diagram of clipboard paste processing 750 according to another embodiment of the invention. The clipboard paste processing 750 pertains to processing associated with a paste command. The clipboard paste processing 750 reflects processing associated with a paste command which follows a previously processed copy command by the clipboard copy processing 700.

[0045] The clipboard paste processing 750 initially opens 752 a second application program. The second application program is also referred to as a destination application program. Next, a decision 754 determines whether a paste command has been received. When the decision 754 determines that a paste command has not been received, then other processing 756 is potentially performed. Following the other processing 756, if any, the clipboard paste processing 750 returns to repeat the decision 754.

[0046] Once the decision 754 determines that a paste command has been received, a decision 758 determines whether the source is secure. The source, for example, can refer to the first application program, the source file or the selected content. When the decision 758 determines that the source is secure, then a decision 760 determines whether the destination is secure. When the decision 760 determines that the destination is secure, then the content from the clipboard buffer can be pasted 762 into the second application program. Here, in the case in which both the source and the destination are deemed secure, the selected content is permitted to be pasted from the clipboard buffer into the second application program that has requested such content.

[0047] However, when the decision 760 determines that the destination is not secure, then alternate content can be pasted 764 into the second application program. As noted above, the alternate content can take a variety of different forms depending upon implementation. Still further, when the decision 758 determines that the source is not secure, then the decision 760 can be bypassed such that the content is pasted 762 from the clipboard buffer into the second

application program. Following the operations 762 or 764, the clipboard paste processing 750 is complete and ends.

[0048] According to one implementation, the decision 758 can utilize a flag or other indicator that may have been stored by the clipboard copy processing 700 in determining whether the source is secure. The destination can be deemed secure if security is imposed, such as by a password or other authentication technique.

[0049] According to another implementation, the destination would be deemed unsecured unless it was the same application as the first application. As an example, a program handler returned from an operating system call (e.g. WIN32 APIs) to retrieve a source file can be compared to another program handler of a second application. If the program handlers match or they are the same program handler, then the source and destination are the same application, and thus the pasting of the selected content into the application program would be permitted. However, if the destination application was not the same as the source application, then, in this embodiment, the pasting of the content would be not permitted (instead, alternate content could be provided).

[0050] Alternatively, in Windows OS, the "active window" or "foreground window" mechanism can be utilized to determine if the selected content in a secured source file is being copied into another application. Although several windows (assuming an application activating a window) may be visible on the screen simultaneously, there is only one window that is active or receives an input from the keyboard or the mouse. By calling an appropriate API (e.g. GetActiveWindowTitle), the clipboard security monitor can be informed if the selected content has been copied into the clipboard or a second application has been activated to retrieve the selected content from the clipboard.

[0051] FIG. 8 is a block diagram of a third representative copy and paste operation 800 according to one embodiment of the invention. The third representative copy and paste operation 800, for example, pertains to the clipboard copy processing 700 illustrated in FIG. 7A and the clipboard paste processing 750 illustrated in FIG. 7B.

[0052] The procedures performed by the third representative copy and paste operation 800 are as follows. A source application 802 operates to display a source file for a user. The user can then interact with the source application 802 to

select content (selected content) 804. Then, a copy button 806 can be activated to store the selected content 804 to a clipboard buffer 808. Typically, the clipboard buffer 808 is associated with a clipboard application.

[0053] Subsequently, at a destination application 810, the user activates a paste button 812. In response, the destination application 810 requests the contents of the clipboard from the clipboard buffer 808. A clipboard security monitor 814 then (assuming the selected content is known or presumed to be secured) interacts with the clipboard buffer 808 and/or the destination application 810 to determine whether the destination application 810 is a secure application. When the clipboard security monitor 814 determines that the destination application 810 is secure, then the selected content can be permitted to be retrieved from the clipboard buffer 808 and pasted (i.e., inserted) into the destination application 810 or a file therein. Additionally, if the clipboard security monitor 814 determines that the source application 802 is not a secure application, then the selected content stored in the clipboard buffer 808 could be supplied to the destination application 810 regardless of whether the destination application was secure or not.

[0054] Most application programs (e.g., Microsoft Word or Notepad) either generate or process data in one form or another. One can readily copy data between various applications using the clipboard provided by the clipboard application. Typically, one copies a portion of data from one application into the clipboard and then pastes the portion of data into other applications as many times as desirable as long as the clipboard has not been updated. The data or content being copied can be text, image, audio or other type data or content.

[0055] The clipboard application is initiated by a user and has a very simple protocol. When the user requests a "copy" or "cut" operation to place data from a first application into the clipboard, the first application makes Application Programming Interface (API) calls to empty the clipboard and then to store the data to the clipboard. In one implementation, the first application provides the single item of data in multiple formats, so as to increase the likelihood that the eventual recipient application will understand one of them. Using this technique allows the first application to simply list the formats it supports for the data, but only has to render the data when another application actually tries to retrieve the data from the

clipboard. In this way, data that is never going to be used never needs to be rendered. When the user makes a paste request, the receiving application enumerates the formats of the data in the clipboard and, having found one it likes, takes the data in that format.

[0056] FIG. 9 illustrates a block diagram of a clipboard usage control system 900 according to one embodiment of the invention. More particularly, the clipboard usage control system 900 provides an exemplary implementation of how the clipboard is blocked to a user of an application accessing a secured document while still being available for other applications accessing unsecured documents.

[0057] The clipboard usage control system 900 includes a document securing module (DSM) 902 that is configured to interface with an operating system (e.g., Microsoft Windows). When a secured document is requested, the document securing module 902 is activated without interaction or notification of the user. In one implementation, the document securing module 902 is similar to a device driver that essentially converts more general input/output instructions of an operating system to messages that a device/module being supported can understand. The device/module enclosed in the document securing module 902 is a cipher having an encryption/decryption process model 904 and a filter function 906. The encryption/decryption process model 904 encrypts a document of secure nature and decrypts a secured document when the secured document is requested. Upon detecting a secured file being accessed, the filter 906 is activated and prevents the clipboard from being used to copy secure content to other applications, particularly other unsecure applications.

[0058] In operation, a user selects a secured document that is associated with an application 908 (e.g., Microsoft Word or PowerPoint). With Microsoft Windows, an operating system (OS) access known as the ProcessID property can be used to activate an application (as an argument to the AppActivate method). The parameter ProcessID identifies the application and an event handler takes necessary parameters to continue the OS access to an Installable File System (IFS) Manager 910. The IFS Manager 910 is responsible for arbitrating access to different file system components. In particular, the IFS Manager 910 is structured as an ordinary Dynamic Link Library (DLL) with entry points for opening, closing,

reading, writing files and others. With one or more flags or parameters (e.g., a file key) passed along, the access activates the document securing module 902.

[0059] According to one embodiment, the document securing module 902 resides on a local disk in a file that is structured like a Dynamically Linked Library (DLL), typically with a SYS or IFS extension, and is loaded during system initialization. Once the document securing module 902 is installed and initialized, a kernel communicates with it in terms of logical requests for file opens, reads, writes, seeks, closes, and so on. Through the IFS Manager 910, a File System Device (FSD) 912 translates these requests--using control structures and tables found on the volume itself--into requests for sector reads and writes for which it can call special kernel entry points called File System Helpers (FsHlps). The kernel passes the demands for sector I/O to an appropriate device driver and returns the results (e.g., the requested document) to the FSD 912.

[0060] Upon receiving the results from the FSD 912 indicating that the requested document is secured, document securing module 902 activates a cipher (e.g., an encryption/decryption module 904) included therein to decrypt the document. At the same time, the filter function 906 can be activated. According to one implementation, an API for the clipboard is called to virtually place a filter in front of (or in back of) the clipboard so that copied data can be restricted as to its use. The appropriate API is available from Microsoft Corporation, namely documentation pertaining to Platform Form Release, Feb, 2001, which is incorporated herein by reference.

[0061] On the source application side, the secured document is opened and perhaps the clear contents therein are displayed in the application. A portion of the clear contents may be selected (i.e., highlighted) for the purpose of copying into another application. When a "copy" command is initiated, the selected contents are attempted to be saved into the clipboard. The filter imposed by the filter function 906 can become effective to void the selected contents. Depending on implementation, the filter may perform differently. In one case, the filter simply blocks any copied data from being placed into the clipboard. The result is that the clipboard is empty and a user cannot transport the copied contents into another application. In another case, the filter can be configured to include a message (e.g., unauthorized copy is not permitted). When a user attempts to copy a

selected portion into another application, the result is that the message (not the selected portion) is copied. In still another case, the clipboard is copied into an illegible or scrambled content corresponding to the selected portion. This, for example, can be done through a simple logical operation on the selected portion.

[0062] FIG. 10 shows some of the possible outcomes of a copy and paste operation from a secured document. As shown in FIG. 10, a window 1000 pertaining to a first application (i.e., Microsoft Windows) displays a secured document. A window 1002 represents the window 1000 after a user has selected a portion of the secured document to be copied. Following a paste command with respect to a destination document of a second application, a window 1004, 1006 or 1008 can be displayed. The window 1004 displays a message indicating that the selected portion is not permitted to be copied. The window 1006 displays a blank area as an indication that the selected portion is not able to be copied. The window 1008 displays scrambled content representing the selected portion but such is not discernable by the user.

[0063] Alternatively, when the selected portion is copied into the clipboard, the selected portion may be modified invisibly to the user. According to one embodiment, one or more special marks may be inserted into the spaces of the selected portion. These special marks are ignorant to an application. For example, a pair of Hexadecimal numbers "FF" can be inserted into the spaces between words or replace the spaces (e.g. "32" in ASCII) while the newly inserted "FF" are neither visible on a display nor printable by a printer. Even if a user was successful in copying the selected portion into the clipboard and pasting the selected portion from the clipboard to a new application, these special marks would go along with the file under the new application with the special marks invisible to users or many utility applications (e.g. printing, display or audio). As a result, it is now possible to trace down where the copied portion in a new file is original from.

[0064] The "cut" command performs a copy command and then deletes the selected content from the source document. Hence, for purposes of storing selected content to a clipboard, a cut command operates in the same manner as the above-discussed copy command.

[0065] In one embodiment, a secured file or secured document can include two parts, an attachment, referred to as a header, and an encrypted document.

The header may include information regarding a file key or the file key itself that can be used to decrypt the encrypted document. The invention is preferably implemented in software, but can be implemented in hardware or a combination of hardware and software. The invention can also be embodied as computer readable code on a computer readable medium. The computer readable medium is any data storage device that can store data which can thereafter be read by a computer system. Examples of the computer readable medium include read-only memory, random-access memory, CD-ROMs, magnetic tape, optical data storage devices, and carrier waves. The computer readable medium can also be distributed over network-coupled computer systems so that the computer readable code is stored and/or executed in a distributed fashion.

[0066] The advantages of the invention are numerous. Different embodiments or implementations may yield one or more of the following advantages. One advantage of the invention is that secured documents or files being displayed are protected from having their content copied to other unsecure applications or documents. Another advantage of the invention is that a clipboard security monitor (or a document securing module) is embedded in an operating system as such the management of the clipboard is transparent to a user. Still another advantage of the invention is that a copy application is reliably managed such that secured content is not able to be copied to unsecure destinations. Still another advantage of the invention is that alternate content can be used in place of secure content that is not permitted to be copied to unsecure destinations.

[0067] The many features and advantages of the present invention are apparent from the written description and, thus, it is intended by the appended claims to cover all such features and advantages of the invention. Further, since numerous modifications and changes will readily occur to those skilled in the art, it is not desired to limit the invention to the exact construction and operation as illustrated and described. Hence, all suitable modifications and equivalents may be resorted to as falling within the scope of the invention.

What is claimed is: